

4

ARTÍCULO DE INVESTIGACIÓN

IMPLEMENTACIÓN DE ALGORITMOS DE COMPUTACIÓN CUÁNTICA Y REDES NEURONALES CONVOLUCIONALES PARA EL RECONOCIMIENTO DE IMÁGENES

IMPLEMENTATION OF QUANTUM COMPUTING ALGORITHMS AND CONVOLUTIONAL
NEURAL NETWORKS FOR IMAGE RECOGNITION

Fecha de recepción: 31/10/2022 | Fecha de aceptación: 23/11/2022

Autor:

¹ **Mendoza Jurado Helmer Fellman**

¹ Universidad Autónoma Juan Misael Saracho

Correspondencia del autor: helmerf.mj7@gmail.com¹,
Tarija- Bolivia.

RESUMEN

La presente investigación busca resolver una necesidad emergente del concepto de Inteligencia Artificial Cuántica, siendo que principalmente busca establecer distintos métodos y modelos de operación basado en algoritmos de la computadora cuántica. El conjunto de algoritmos cuánticos implementados en este proceso de investigación permiten procesar una imagen clásica en un estado cuántico, esto fundamentalmente para poder seleccionar límites y permitir la conversión de una imagen en una escala de monocromática binaria, tal cual expresa la lógica y metodología de una red neuronal convolucional (CNN), lo cual permite establecer distintas posibilidades que se fundamentan en la teoría de la información cuántica de interpretación de problemas clásicos de las Ciencias Computacionales, por tanto, el objetivo del presente trabajo es implementar un conjunto de algoritmos cuánticos que permita resolver el problema de reconocimiento de imágenes utilizando métodos y características de la computación cuántica, evaluando la correcta implementación algoritmos de reconocimiento de patrones existentes y creando un modelo de reconocimiento efectivo utilizando las características y métodos de la computación cuántica. La relevancia de esta investigación radica en el modelado matemático y la implementación de un conjunto de notebooks de algoritmos cuánticos para resolver varios de los problemas clásicos. La novedad científica de esta área se expresa principalmente en la constante actualización y adición del campo de la investigación cuántica en una serie de áreas, y la simulación por computadora de los fenómenos y características de la física cuántica no está suficientemente investigada en el mundo. Actualmente, en muchos países, se está llevando a cabo una investigación intensiva sobre el desarrollo y la creación de computadoras cuánticas y su software, por tanto, el presente artículo representa rigurosamente algunos resultados teóricos y prácticos en el campo de la computación cuántica y su modelo matemático.

ABSTRACT

This research seeks to solve an emerging need for the concept of Quantum Artificial Intelligence, mainly seeking to establish different methods and models of operation based on quantum computer algorithms. The set of quantum algorithms implemented in this research process allow processing a classical image in a quantum state, this is fundamentally to be able to select limits and allow the conversion of an image into a binary monochrome scale, as it expresses the logic and methodology of a convolutional neural network (CNN), which allows establishing different possibilities that are based on the theory of quantum information for the interpretation of classical problems of Computer Science, therefore, the objective of this work is to implement a set of quantum algorithms that allows solving the image recognition problem using quantum computing methods and features, evaluating the correct implementation of existing pattern recognition algorithms, and creating an effective recognition model using quantum computing features and methods. The relevance of this research lies in the mathematical modeling and implementation of a set of quantum algorithm notebooks to solve several of the classical problems. The scientific novelty of this area is mainly expressed in the constant updating and addition of the field of quantum research in a number of areas, and the computer simulation of quantum physics phenomena and characteristics is not sufficiently researched in the world. Currently, in many countries, intensive research is being carried out on the development and creation of quantum computers and their software, therefore, the present article rigorously represents some theoretical and practical results in the field of quantum computing and its model. mathematical.

Palabras Claves: Qubit, Entrelazamiento, Superposición, Circuito cuántico, Computa puerta cuántica

Keywords: Qubit, Entanglement, Superposition, Quantum Circuit, Quantum Gate.

1. INTRODUCCIÓN

En la actualidad las Ciencias Computacionales están trabajando activamente en el estudio y la implementación física de una computadora cuántica. En la actualidad se han construido varios prototipos de dispositivos informáticos cuántico en diferentes momentos y distintas partes del mundo, principalmente desde un enfoque que busca establecer de manera rigurosa y metodológicamente correcta todas las características esenciales de la mecánica cuántica (entrelazamiento, superposición y reversibilidad).

El campo de los algoritmos cuánticos se actualiza y complementa constantemente, sin embargo, durante mucho tiempo no existieron lenguajes de programación compatibles con el concepto de Qubit (Quantum bit), para propósitos y tareas prácticas, siendo que el resultado del algoritmo cuántico es probabilístico, y de ahí se fundamenta un pequeño incremento en el número de operaciones en el algoritmo y la maximización del entrelazamiento de Qubits, puede llevar arbitrariamente la probabilidad de obtener el resultado correcto a la unidad.

La implementación práctica de estas tareas en sistemas modernos requiere un período de tiempo inaceptablemente largo. Recientemente, se estableció un rápido aumento del interés en las computadoras cuánticas (Boixo et al., 2016, p. 2)

Principalmente debido a que los métodos modernos para reconocer objetos tienen una serie de limitaciones importantes: errores de búsqueda en grandes bases de datos y determinación de un objeto al cambiar su posición, deterioro en la calidad de identificación de objetos en función de la calidad de la iluminación y herramientas de disfraz.

Se supone que utiliza algoritmos (Stierhoff & Davis, 1998, pp. 29–35) de naturaleza cuántica para determinar objetos e imágenes.

El uso de computadoras cuánticas puede aumentar significativamente la velocidad de resolución de problemas computacionales (Lipschutz et al., 2009)

y, lo más importante, aumentar exponencialmente la velocidad de resolución de problemas NP-completos que pueden resolverse en un tiempo inaceptable en las computadoras clásicas.

2. MATERIALES Y MÉTODOS

El proceso de investigación fue desarrollado desde un enfoque cuantitativo, fundamentalmente desde la gestión de imágenes que serán procesadas por una red neuronal convolucional, que intuitivamente se integra al modelo de predicción propuesto, siendo fundamentado por un paradigma positivista, así mismo el proceso de investigación subyace desde un tipo de investigación observacional y descriptivo, principalmente por la rigurosidad metodológica que implica la implementación de modelos de Deep Learning y la base algorítmicamente de la computación cuántica, que se especifica posteriormente.

2.1. MODELADO DE UNA CLASE DE IMÁGENES FACIALES UTILIZANDO EL ANÁLISIS DE COMPONENTES PRINCIPALES

El método de componentes principales es uno de los métodos más comunes para reducir la dimensionalidad de los datos y la pérdida de la menor cantidad de información. El método permite resaltar los rasgos característicos del rostro y utilizarlos para la reconstrucción y recuperación. Este método se basa en la transformación de Karunen-Loev.

La idea principal del método es representar imágenes de rostros de personas en forma de un conjunto de componentes principales de imágenes llamados "rostros propios".

Un atributo útil de las caras propias es que la imagen correspondiente a cada componente tiene una forma similar a una cara, como se muestra en la Figura 1.

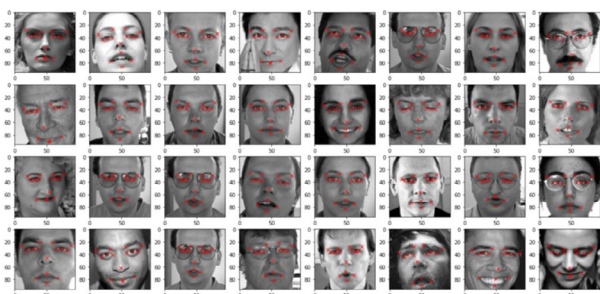


Figura 1, imágenes denominadas como "Rostros propios"
Fuente: Elaboración propia de acuerdo a un dataset de rostros descargado de Kaggle.

El cálculo de los componentes principales se basa en el cálculo de los autovectores y autovalores de la matriz de covarianza (Collier, 1993), que parte de calcular resultados significativos a partir de la imagen propiamente dicha, estableciendo que la suma de los componentes principales multiplicada por los vectores propios correspondientes es una reconstrucción de la imagen.

El principio básico de los programas de rastreo y reconocimiento facial funciona de la siguiente manera, en el primer paso, se analiza la imagen, posteriormente se buscan los rostros, asimismo se establece el procesamiento destinado a resaltar las características individuales de cada rostro humano detectado, a fin de que, en función de las características individuales seleccionadas, se compila una plantilla digital. Las características individuales en este caso son la distancia entre los ojos, la profundidad de su ajuste, la forma de los pómulos, la nariz y la mandíbula.

Todo esto se convierte en un código digital, que es algo similar a un "elenco facial digital", que luego se podría utilizar para comparar, por ejemplo, con los delincuentes buscados. En la mayoría de los casos, es casi imposible restaurar la imagen inversa. Este enfoque es la base de la biometría.

2.2. MODELADO DE UNA CLASE DE IMÁGENES FACIALES MEDIANTE ANÁLISIS FACTORIAL

El Análisis Factorial (AF), como muchos métodos de análisis de datos multidimensionales, se basa en la hipótesis de que las variables observadas son manifestaciones indirectas de un número relativamen-

te pequeño de algunos factores ocultos. El análisis factorial, por lo tanto, es una combinación de modelos y métodos enfocados en identificar y analizar relaciones ocultas (latentes) entre las variables observadas. En el contexto de los problemas de reconocimiento, las variables observables suelen ser signos de objetos. FA puede considerarse como una generalización del método de componentes principales. El propósito del análisis factorial en el contexto del problema de detección de rostros es obtener un modelo de una imagen de rostro (con un número previsible de parámetros), con el que se puede evaluar la proximidad de una imagen de prueba a una imagen de rostro.

2.3. MÁQUINAS DE SOPORTE VECTORIAL (SVM)

El objetivo de entrenar a la mayoría de los clasificadores es minimizar el error de clasificación en el conjunto de entrenamiento (llamado riesgo empírico). Por el contrario, utilizando el método del vector de soporte, se puede construir un clasificador que minimice la estimación superior del error de clasificación esperado. La aplicación del método del vector de soporte al problema de detección de rostros consiste en la búsqueda de un hiperplano en el espacio de características, que separa la clase de imágenes faciales de las imágenes no faciales. La posibilidad de una separación lineal de clases tan complejas como imágenes de rostros y "no rostros" parece poco probable. Sin embargo, la clasificación mediante vectores de soporte hace posible utilizar el aparato de funciones nucleares para proyectar implícitamente vectores de características en un espacio de dimensión potencialmente mucho mayor (incluso mayor que el espacio de la imagen), en el que las clases pueden ser linealmente separables.

2.4. REDES NEURONALES CONVOLUCIONALES (CNN)

Las redes neuronales (Kovera, 2018, pp. 73–74) se han utilizado durante mucho tiempo y con éxito para resolver muchos problemas de reconocimiento. Para resolver el problema de la detección de rostros, se utilizaron una gran cantidad de redes neuronales

de diversas arquitecturas. Una ventaja de utilizar redes neuronales para resolver el problema de la detección de rostros es la capacidad de obtener un clasificador que permita modelar la función de distribución compleja de las imágenes faciales. Siendo que una desventaja significativa que subyace en la necesidad de un ajuste metódico y minucioso de la red neuronal para obtener un resultado de clasificación eficiente.

2.5. MODELOS OCULTOS DE MARKOV (HMM)

Los modelos ocultos de Markov (HMM) es una de las formas de obtener un modelo matemático (descripción de las propiedades) de alguna señal observable. Los modelos ocultos de Markov pertenecen a la clase de modelos estocásticos, que intentan caracterizar solo las propiedades estadísticas de la señal, sin tener información sobre sus propiedades específicas.

La base de los modelos estocásticos es la suposición de que la señal puede describirse mediante algún proceso aleatorio paramétrico y que los parámetros de este proceso pueden estimarse con bastante precisión de una manera completamente definida. Los modelos personalizados de Hidden Markov pueden considerarse como una fuente de alguna señal aleatoria con características bien definidas. Para una MMC sintonizada, es posible calcular la probabilidad de generar una señal de prueba con este modelo.

Como una aplicación al problema de reconocimiento, al presentar el vector de características de un objeto en forma de señal (un conjunto de observaciones consecutivas), puede simular una clase de objetos usando el CMM.

La probabilidad de que el objeto de prueba pertenezca a la clase especificada por el CMM se estima como la probabilidad de generar una señal correspondiente a su vector de características.

Configurar (entrenar) CMM - consiste en modificar sus parámetros para lograr la máxima probabilidad de generar señales correspondientes a los vectores del conjunto de entrenamiento.

3. RESULTADOS

3.1. EL PROCESO DE FORMACIÓN DE UN CONJUNTO DE QUBITS

En la Figura 2 se muestra el proceso computacional (Olukotun et al., 2007) de modelado de cálculos cuánticos entrelazados (Moran, 2019, pp. 42–43) en el campo de los algoritmos cuánticos. Implica principalmente el uso de varios tipos de cálculos cuánticos de algoritmos cuánticos y la formación de un conjunto de qubits para el estado de las señales de control de normalización en un punto particular en el tiempo.

```
# Ejecución de qubits básicos.
ax=plot_bloch(zero_qubit,color='xkcd:red')
plot_bloch(one_qubit,color='xkcd:orange',ax=ax)
plot_bloch(plus_qubit,color='xkcd:yellow',ax=ax)
plot_bloch(minus_qubit,color='xkcd:green',ax=ax)
plot_bloch(clockwisearrow_qubit,color='xkcd:blue',ax=ax)
plot_bloch(counterclockwisearrow_qubit,color='xkcd:purple',ax=ax)

# Qubit desde 10% |"0"> y 90% |"1">
plot_bloch(zero_to_one_qubit(10,90),color="xkcd:turquoise",ax=ax)
```

<matplotlib.axes._subplots.Axes3DSubplot at 0x1113c5710>

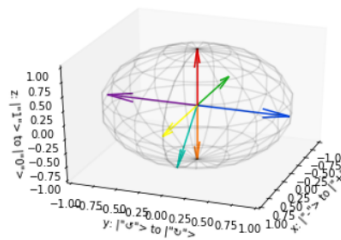


Figura 2: proceso computacional de cálculos cuánticos entrelazados. Fuente: Elaboración propia del código en Python en un notebook de Jupiter Notebook.

Aplicando el producto tensorial entre transformadas de Hadamard (Viktor Potapov et al., 2018, p. 199), obtenemos términos de la forma $K_p^n \otimes K_p^{(n2)}$ y combinaciones similares de factores de ganancia. El ejemplo descrito anteriormente indica la existencia de dieciséis estados probabilísticos que definen variaciones entre correlaciones según su tipo y tipo. Consideremos el método cuántico, destinado a la presentación y procesamiento de fotografías o imágenes de píxeles en color. Se supone que cada píxel de la imagen $x(i,j)$ se transforma en un cierto estado cuántico $|q(i,j)\rangle: |q(i,j)\rangle = c_0 |0\rangle + c_1 |1\rangle$. Codificaremos la paleta de colores del conjunto de píxeles en el marco de estados cuánticos de amplitud compleja:

$$\delta: R^3 \rightarrow C_1^2, (x_1, x_2, x_3) \mapsto (r_1 e^{i\phi_1}, r_1 e^{i\phi_2})$$

Formula 1

Donde x_1, x_2, x_3 son los valores del sistema de color RGB, $r_1 = \sqrt{1-x_3^2}, r_2 = x_3, \phi_1 = \arcsin(2x_2-1), \phi_2 = \arcsin(2x_2-1)$, luego obtenemos un conjunto de colores de píxeles en la forma $|q\rangle$. El siguiente paso es el proceso de codificación de las coordenadas de la paleta de colores del conjunto de píxeles:

$$|k\rangle = |x\rangle|y\rangle = |x_{n-1}, x_{n-2} \dots x_0\rangle|y_{n-1}, y_{n-2} \dots y_0\rangle, x_i, y_i \in \{0,1\}$$

Formula 2

Donde, $|x\rangle, |y\rangle$ son estados cuánticos que codifican una cuadrícula de coordenadas de píxeles.

Dependiendo del algoritmo de transformación utilizado, la selección de valores de probabilidad de amplitud que codifican los colores de píxeles de una fotografía puede variar significativamente. Luego crea una superposición del proceso de cálculo. Superposición de píxeles de estados de un sistema cuántico de una fotografía o imagen en la entrada:

$$|I\rangle = \frac{1}{2^n} \sum_{k=0}^{2^{2n}-1} |q_k\rangle \otimes |k\rangle$$

Formula 3

Etapa 1. Tengamos una imagen de dimensión con un conjunto de píxeles en escala de grises (Boneh & Zhandry, 2013). Convertiremos el conjunto completo de píxeles de la foto $x(i, j)$ ingresando la entrada en el estado $|q(i, j)\rangle$ del sistema cuántico. Con la intensidad de todo el conjunto de píxeles, las distribuciones de probabilidad $|c_0|^2$ y $|c_1|^2$ se calculan determinando las sumas y de la siguiente manera:

$$s_1 = \sum_{l=0}^1 x(i-k, j-l) + x(i, j+l) + x(i, j+2)$$

Formula 4

$$s_2 = x(i-1, j-1) + x(i-1, j) + x(i, j-1)$$

Formula 5

Si $P=(s1+s2)/5$, entonces, $|c0|^2=f(P), |c1|^2=1-f(P)$, donde $f(P)=1/(1+e^{(P+a)/b})$. El estado cuántico $|0\rangle, |1\rangle$, corresponden a los vectores $(1/0), (0/1)$. De acuerdo con este qubit $|q(i, j)\rangle$, el vector $((1-f(P))/f(P))$, está mapeado, por lo que la intensidad del conjunto de píxeles de la fotografía se representa en un espacio bidimensional.

Etapa 2. Al medir el qubit $|q(i, j)\rangle$ de todo el conjunto de píxeles de la foto alimentada a la entrada, se forma la matriz del sistema de píxeles. Si los estados cuánticos $|0\rangle, |1\rangle$ corresponden a los indicadores 0, 1 del conjunto de píxeles de salida de la foto, obtenemos un objeto binario en la salida.

3.2. MÓDULOS DE EMULACIÓN DE ALGORITMOS CUÁNTICOS Y PROGRAMACIÓN

Inicialmente se implementó Python como lenguaje de programación cuántica el cual fue desarrollado con el consumo de un servicio API de la computadora cuántica "IBM QX", siendo que la interfaz general del modelo desarrollado para un circuito cuántico reversible se muestra en la Figura 3.

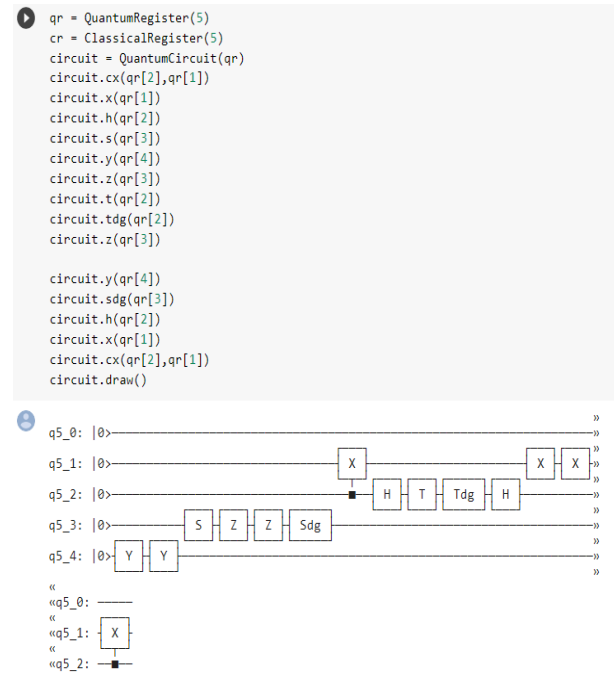


Figura 3. Circuito Cuántico Reversible.
Fuente: Elaboración propia del código en Python en un notebook de Jupiter Notebook.

En la esquina inferior izquierda hay botones de los módulos ilustrados arriba: emulaciones de algoritmos cuánticos ("Figura 4 con el algoritmo de Shor", "Figura 5 con el algoritmo de Grover", "Figura 4 con el algoritmo de QFT) y lenguajes de programación cuántica ("Open QCL").

```
def shors_algorithm_quantum(N, fixed_a=None):
    assert(N>0)
    assert(int(N)==N)
    while True:
        if not fixed_a:
            a=random.randint(0,N-1)
        else:
            a=fixed_a
        g=math.gcd(a,N)
        if g!=1 or N==1:
            first_factor=g
            second_factor=int(N/g)
            return first_factor,second_factor
        else:
            r=period_finding_quantum(a,N)
            if not r:
                continue
            if r % 2 != 0:
                continue
            elif a**(int(r/2)) % N == -1 % N:
                continue
            else:
                first_factor=math.gcd(a**int(r/2)+1,N)
                second_factor=math.gcd(a**int(r/2)-1,N)
                if first_factor==N or second_factor==N:
                    continue
                if first_factor*second_factor!=N:
                    # checking our work
                    continue
                return first_factor,second_factor

# Here's our final result
shors_algorithm_quantum(15, fixed_a=2)
```

Figura 4, Algoritmo Cuántico de Shor.
Fuente: Elaboración propia del código en Python en un notebook de Jupiter Notebook.

```
import math
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
qr = QuantumRegister(3)
cr = ClassicalRegister(3)
circuit = QuantumCircuit(qr, cr)
circuit.h(qr[0])
circuit.cu1(math.pi/2.,qr[1],qr[0])
circuit.cu1(math.pi/4.,qr[2],qr[0])
circuit.h(qr[1])
circuit.cu1(math.pi/2.,qr[2],qr[1])
circuit.h(qr[2])
circuit.draw()
```

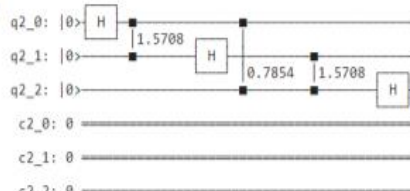


Figura 5, Algoritmo de la transformada Cuántica de Fourier en tres Qubits.
Fuente: Elaboración propia del código en Python en un notebook de Jupiter Notebook.

```
def grovers_algorithm(checker,num_inputs,num_registers,num_iterations=None):
    if num_iterations == None:
        from math import floor,sqrt
        iterations=floor(sqrt(2**num_inputs))
    else:
        iterations=num_iterations
    print("Running Grover's algorithm for %d iterations"%iterations)
    qr = QuantumRegister(num_registers)
    cr = ClassicalRegister(num_registers)
    qc = QuantumCircuit(qr,cr)
    # Configuring the input
    for i in range(num_inputs):
        qc.h(qr[i])
    # Setting up the output of the checker function
    qc.x(qr[num_registers-1])
    qc.h(qr[num_registers-1])
    # Do the Grover's steps
    for it in range(iterations):
        checker(qr,qc)
        mover(qr,qc,num_inputs)
    # Measure the inputs
    for j in range(num_inputs):
        qc.measure(qr[j], cr[j])
    return cr,qr,qc
```

Figura 6: Algoritmo Cuántico de Grovers
Fuente: Elaboración propia del código en Python en un notebook de Jupiter Notebook.

Para comenzar es necesario establecer que los algoritmos desarrollados con MQC, deben inicializar en el circuito, que por inercia permite que exista varios tipos de datos cuántico básico Qureg (registro cuántico, registro cuántico). Puede representarse como una matriz de Qubits.

4. DISCUSIÓN

En la actualidad se puede establecer que el desarrollo tecnológico de las Ciencias es fundamentalmente a la incursión avanzada que la sociedad científica mundial refiere a la implementación de modelos fundamentados en la Inteligencia Artificial, y principalmente en el desarrollo de productos y servicios que cuentan de manera nativa con modelos de Redes Neuronales Convolucionales, desde vehículos autónomos, hasta el servicio de reconocimiento facial, más aún con la investigación de un modelo contemporáneo computadora que cumpla implícitamente las características fundamentales de la mecánica cuántica, el hecho de usar estas tecnologías transversalmente establece al mejor de los panoramas en ciencia e investigación para poder expandir potencialmente los beneficios que indiscutiblemente puede ofrecer favorablemente en distintas área de la investigación para la sociedad mundial.

5. CONCLUSIÓN

Durante la redacción de este trabajo, se realizó un análisis de los principios básicos para la implementación de algoritmos en el marco de modelos de computadoras cuánticas, siendo que se distinguen los algoritmos cuánticos básicos, se describen sus implementaciones teóricas y de software. Se deriva una dependencia directa del tiempo de funcionamiento de una transformación cuántica particular (algoritmo) del número de Qubits asignados a esta transformación. Se muestra la implementación tanto de los módulos para emular algoritmos cuánticos como del entorno para el desarrollo de la computación cuántica (programación cuántica) y el modelo completo de la computadora cuántica en su conjunto. El deseo de incrementar la potencia informática de las computadoras y proporcionar una escala de tareas insuperable es uno de los factores determinantes en el desarrollo de las tecnologías de supercomputadoras. Las principales ventajas de utilizar la computación cuántica en el campo de la determinación de objetos e imágenes: aceleración del proceso de computación usando componentes cuánticos, estabilidad en diferentes ángulos del objeto, su movimiento y estática, asegurando inmunidad al ruido criptográfico (Victor Potapov et al., 2016). Se concede gran importancia al desarrollo de principios físicos fundamentalmente nuevos de la computación, donde la computación cuántica es el área más prometedora.

6. REFERENCIAS BIBLIOGRÁFICAS

- Boixo, S., Isakov, S. V., Smelyanskiy, V. N., Babbush, R., Ding, N., Jiang, Z., Bremner, M. J., Martinis, J. M., & Neven, H. (2016). Characterizing Quantum Supremacy in Near-Term Devices. <https://doi.org/10.1038/s41567-018-0124-x>
- Boneh, D., & Zhandry, M. (2013). Quantum-secure message authentication codes. Annual International Conference on the Theory and Applications of Cryptographic Techniques, 592–608.
- Collier, D. (1993). The comparative method. Political Science: The State of Discipline II, Ada W. Finifter, ed., American Political Science Association.
- Kovera, A. (2018). How to Create Machine Superintelligence: A Quick Journey Through Classical/Quantum Computing, Artificial Intelligence, Machine Learning, and Neural Networks (Second Edition). CreateSpace Independent Publishing Platform. <https://books.google.com/books?id=4b8jtQEACAAJ>
- Lipschutz, S., Lipson, M. L., York, N., San, C., Lisbon, F., Madrid, L., City, M., New, M., San, D., Singapore, J. S., & Toronto, S. (2009). SCHAUM'S outlines Linear Algebra Fourth Edition Schaum's Outline Series.
- Moran, C. C. (2019). Mastering Quantum Computing with IBM QX: Explore the world of quantum computing using the Quantum Composer and Qiskit. Packt Publishing. <https://books.google.com/books?id=jCWGDwAAQBAJ>
- Olukotun, K., Hammond, L., & Laudon, J. (2007). Chip multiprocessor architecture: techniques to improve throughput and latency. Synthesis Lectures on Computer Architecture, 2(1), 1–145.
- Potapov, Victor, Gushansky, S., Guzik, V., & Polenov, M. (2016). Architecture and software implementation of a quantum computer model. Computer Science On-line Conference, 59–68.
- Potapov, Viktor, Gushanskiy, S., Guzik, V., & Polenov, M. (2018). The computational structure of the quantum computer simulator and its performance evaluation. Computer Science On-line Conference, 198–207.
- Stierhoff, G. C., & Davis, A. G. (1998). A history of the IBM Systems Journal. IEEE Annals of the History of Computing, 20(1), 29–35. <https://doi.org/10.1109/85.646206>